Statistic Multiplexed Computing

-- A Position Paper on Operating System and Software for Exascale Systems

Justin Y. Shi | shi@temple.edu

Abstract

Dataflow parallel processing has earned its inefficiency fame during early unsuccessful trials. The HPC community has since departed from implicit parallelism. Explicit parallel programming paradigms, such as MPI and OpenMP have made great strides to demonstrate the capabilities of supercomputers. They have also left one runaway challenge: **performance scalability**. It has become widely accepted that to get higher performance, reliability must be sacrificed. To get higher reliability, performance must be sacrificed. As for all scientific and engineering development efforts, these seemingly insurmountable difficulties indicate that we are probably suffering from a structural (architectural) problem.

This position paper suggests a different direction: statistic multiplexed computing by ***eliminating the reliable application-level communication assumption in all API's***. Our proposal is based on a practical observation: applications fail at the application programming interface (API) level due to the lack of timeout discipline. Our layered architecture discipline: application, operating system, communication stack, and device driver have semantic gaps that are responsible for the scalability difficulties. Specifically, the problem seems located exactly at the handoff point between the communication stack (TCP/IP) and application API. A running application will fail if its remote communication partner crashes before completing the expected functions, even if all data packets are transmitted correctly.

Statistic multiplexed computing is a natural solution for arbitrary application crashes. Its "cousin" – packet switching technology -- has been successfully used in networks to build the Internet today. Dataflow parallel programming paradigm is a natural fit for statistic multiplexed computing. Although counter-intuitive, its technical depth ensures the eventual success in exacscale systems and for big data processing at the same time.

Challenges Addressed: Scalability of application performance (speed) and reliability for compute intensive and data intensive applications.

Maturity: We have running prototype (at Temple University http://spartan.temple.edu/synergy) for compute intensive applications and a commercial product (at Parallel Computers Technology Inc. http://www.pcticorp.com) for transaction processing systems (SQL Server only). The data intensive system has been sold worldwide under the name of DB^x since 2006.

Uniqueness: Use of statistic multiplexing to solve arbitrary message loss problem was proposed five decades ago. The use of statistic multiplexing to solve computing loss is unique.

Novelty: The novelty of the proposed method lies in its ability to reverse the inherent negative performance and reliability trends in deterministic architectures. Incidentally, we have also found that the assumption of reliable application-level communication is the culprit.

Effort: The Synergy data parallel programming system is just a prototype for demonstration purposes. More development work is necessary to make it widely deployed for practical scientific computing project. Depending upon funding, it is possible to ramp up the system for community test and adaptation within

one year time frame. The DBx system only supports Microsoft SQL Server protocol (TDS). Its core technology can be ported to support exac-scale file systems, different transaction processing engines (Oracle, DB2, Postgress and other relational database systems) and non-SQL (Casandra, StreamBase, etc.). The efforts are dependent on the scope of interest.

References:

1. G. Gibson, "Reflections on failure in post-terascale parallel computing,"in Proceedings of 2007 International Conference on Parallel Processing,2007.
2. A. Fekete, N. Lynch, Y. Mansour, and J. Spinelli, "The impossibility of implementing reliable communication in the face of crashes," J.ACM, vol. 40, pp. 1087–1107, November 1993. [Online]. Available: http://doi.acm.org/10.1145/174147.169676
3. Gilbert and Lynch, "Brewers conjecture and the feasibility of consistent, available, partition-tolerant web services," in ACM SIGACT News (2002), vol. 33(2). ACM, 2002, p. 59.
4. J. Y. Shi, M. Taifi, A. Khreishah, and J. Wu, "Tuple switching network – when slower maybe better," International Journal of Parallel and Distributed Computing, 2011.
5. J. Y. Shi, "A distributed programming model and its applications to computation intensive applications for heterogeneous environments," in International Space Year Conference on Earth and Space Information Systems, Pasadena, CA., February 1992, pp. 10–13.
6. J. B. Dennis, "Data flow supercomputers," Computer, vol. 13, no. 1, pp. 48–56, 1980.
7. M. Taifi, A. Khreishah, and J. Y. Shi, "Natural hpc substrate: Exploitation of mixed multicore cpu and gpus," in HPCS 2011. IEEE, 2011.
8. J. Y. Shi, "Heterogeneous computing for graphics applications," in National Conference on Graphics Applications, April 1991.
9. J. Y. Shi, M. Taifi, A. Khreishah, A. Predeep, and V. Anthony, "Cloud or cluster? - program scalability analysis of hpc benchmarks," Technical Report, CIS Department, Temple University 2012.
10. S. S. Shim, "The cap theorem's growing impact," IEEE Computer, pp. 21–22, 2012.
11. E. Brewer, "Towards robust distributed systems," in Proceedings of the Annual ACM Symposium on Principles of Distributed Computing. ACM, 2000.
12. J. Y. Shi, "High performance lossless esb architecture with data protection for mission-critical applications," in Computer Science and Information Engineering, 2009 WRI World Conference. IEEE, 2009.
13. J. Gray, "The transaction concept: Virtues and limitations," Tandem TR 81.3, Tandem Computers Inc., 1981.
14. M. Stonebraker, "The case for shared nothing architecture," Database Engineering, vol. 9, no. 1, 1986.
15. J. Y. Shi and SuntainSong, "On the assumptions of cap theorem and the big data challenges," Technical Report, Parallel Computers Technology Inc. 2012.
16. J. Y. Shi, Chapter 19: Fundamentals of cloud application architectures, Cloud computing: methodology, system, and applications. CRC, Taylor & Francis Group, 2011.
17. J. G. et. al., "The dangers of replication and a solution," in ACM SIGMOD International Conference on Management of Data Archive, Montreal, Quebec, Canada, 1996, pp. 173 – 182.
18. S. Song, "Method and apparatus for database fault tolerance with instant transaction replication using off-the-shelf database servers and low bandwidth networks," U.S. Patent #6,421,688, 2002.
19. J. Y. Shi and S. Song, "Apparatus and method of optimizing database clustering with zero transaction loss," U.S. Patent Application: #2008018969, 2008.
20. Y. Shi, Synergy v3.0 Manual, 1996, [online] http://spartan.cis.temple.edu/synergy/.